UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/608,320	06/27/2003	Lih-Jyh Weng	3123-554/MAX-017AUS	7998

7590 11/27/2006

Patricia A. Sheehan
Cesari and McKenna, LLP
88 Black Falcon Ave.
Boston, MA 02210

EXAMINER	
CHAUDRY, MUJTABA M	

ART UNIT	PAPER NUMBER
2133	

DATE MAILED: 11/27/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/608,320	WENG ET AL.	
	Examiner	Art Unit	
	Mujtaba K. Chaudry	2133	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 September 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-3 and 5-27* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-3 and 5-27* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on September 01, 2006 has been entered.

Response to Amendment

Applicant's arguments/amendments with respect to amended claims 1, 3, 5, 12, 20, 21 and 23-27 and previously presented claims 2, 6-11, 13-19 and 22 filed September 01, 2006 have been received. All arguments have been fully considered but are not persuasive. The Examiner would like to point out that this action is made final (See MPEP 706.07a).

Applicant contends, "...Fuoco (prior art of record) does not teach data words are stored in a data buffer location and the parity check bits are stored in parity check location..." The Examiner respectfully disagrees. Fuoco teaches (Figure 1a and col. 3, lines 46-52, for example) a **memory subsystem 22 which comprises a plurality of DRAMs 26 with associated ECC logic 28**. The Examiner would like to point out that the memory stores the data (in the Dram) followed by the check bits in the ECC logic 28 as stated in the present application.

Applicants also contend, "...Fuoco does not teach applying data words as a plurality of data words..." The Examiner respectfully disagrees. Fuoco teaches (abstract, for example), "...and logic to do partial writes of data words. The system has a central processing unit (CPU), a BUS interconnecting the CPU and the add-on memory cards. The CPU or associated components are **configured to write data and read data from the add-on memory as several data bytes constituting data words**.

The Examiner disagrees with the Applicant and maintains rejections with respect amended claims 1, 3, 5, 12, 20, 21 and 23-27 and previously presented claims 2, 6-11, 13-19 and 22. All arguments have been considered. It is the Examiner's conclusion that amended claims 1, 3, 5, 12, 20, 21 and 23-27 and previously presented claims 2, 6-11, 13-19 and 22, as presented, are not patentably distinct or non-obvious over the prior art of record.

Claim Rejections - 35 USC § 103

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

Claim 1-3 and 5-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fuoco et al. (USPN 5452429).

As per claim 1, Fuoco et al. (herein after referred to as one entity: Fuoco) substantially teaches (Figure 1 and abstract) a computer system and method with add-on memory cards that have error correction code logic on the card, and logic to do partial writes of data words. The system has a central processing unit (CPU), a BUS interconnecting the CPU and the add-on memory cards. The CPU or associated components are configured to write data and read data from the add-on memory as several data bytes constituting data words. The system is further configured either within the CPU or as a separate function to generate parity bits associated with each of the bytes of data the CPU writes to the add-on memory and to read parity bits associated with data the CPU reads from the add-on memory and regenerate new parity bits and compare the newly generated parity bits with the original parity bits to detect data errors on data read from the add-on memory. The system itself does not contain error correction code (ECC). The add-on memory has ECC logic to identify any byte having a single bit error in the data bytes or the parity bits written by the CPU to the add-on memory and to correct all single bit errors in data read from the add-on memory to the CPU. The error correcting code includes logic to generate parity bits in the data bytes written by the CPU to the add-on memory and logic to compare the parity bits written by the CPU with those generated by the error correcting code logic.

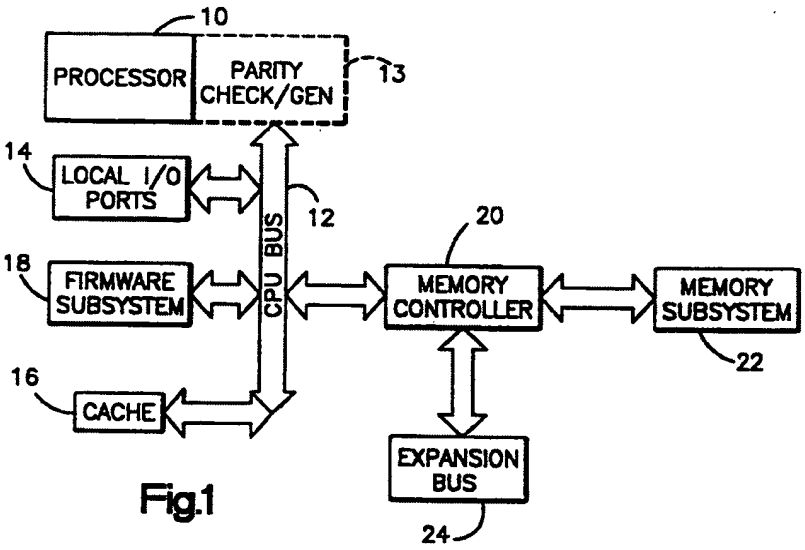

Fig.1

Fuoco does not explicitly teach storing the data and parity check bits on a buffer memory as stated in the present application.

However, Fuoco teaches (col. 10 and Figure 7) the RAS and CAS lines are each directed to the input side of OR-gates 80, 82, 84, 86, 88, and 90. The output of the OR-gates 80, 82, 84, 86, 88, and 90 are directed respectively to a series of non-inverting buffers which are minus active enable. The buffers 92, 94, 96, and 98 have inputs from programmable presence detects on the card and the buffers 100 and 102 have grounded input. It is the output from the buffer 100 which is used to detect whether a card is compatible and the output is supplied to one of pins on the SIMM card. The reason for this configuration is that there are only a limited number of pins on an IBM or other SIMM card, and it is the pin on the card which interacts with the output of buffer 100 which configures the system at all cycles other than read or write or refresh output of buffer 100 will be high. The state will signify the card as either a conventional parity card or a card with ECC on board. Furthermore, the Examiner would like to point out that the buffer

memory is a memory that is between data input and active storage. The data input being the information from the process 10 and the active storage being the memory subsystem 22. Therefore, if would have been obvious to one of ordinary skill in the art at the time the invention was made to check the stored data and parity check bits on a buffer memory. This modification would have been obvious to one of ordinary skill in the art because one of ordinary skill in the art would have recognized that by storing the data and parity check bits on a buffer memory the system would improve on the communication between the processor and active storage by reducing the number of possible transmitted errors.

As per claim 2, Fuoco substantially teaches, in view of above rejections, (Figure 3) the check bits from the check bit generation 44 and the check bits from the memory are XORed 48 and then a syndrome is generated in reference block 46.

As per claim 3, Fuoco substantially teaches, in view of above rejections, (col. 3) the address locations in add-on memory are assumed to be 40 bits wide and the data words are written as 4 byte strings with 7 check bits generated thus accounting for 39 of the possible 40 bits in each address. Such a system is conventional and need not be described further. The 40th bit is used as a flag bit for the syndrome decode.

As per claims 5 and 6, Fuoco substantially teaches, in view of above rejections, (col. 3) upon writing data by the CPU 10, parity bits are generated for each byte of information written to memory by the parity generating and checking device 13 which also checks parity on information read from the memory subsystem 22 during a read cycle to determine parity error. The memory controller also provides the necessary signals, such as Row Activation Strobe (RAS), Column Activation Strobe (CAS), Write Enable (WE), Output Enable (OE), and Address

(ADDR), etc. to the memory subsystem 22 as shown in FIGS. 1A and 1B. The memory controller reads and writes both data and parity to each of the DRAM cards 26, also as shown in FIGS. 1A and 1B.

As per claim 7-11, Fuoco substantially teaches, in view of above rejections, (col. 1) the add-on memory has ECC logic to identify any byte having a single bit error in the data bytes or the parity bits written by the CPU to the add-on memory and to correct all single bit errors in data read from the add-on memory to the CPU. The error correcting code includes logic to generate parity bits in the data bytes written by the CPU to the add-on memory and logic to compare the parity bits written by the CPU with those generated by the error correcting code logic.

As per claim 12-22, Fuoco substantially teaches, in view of above rejections, (cols. 4-6 and Table 1) the participating data bits are labeled from 0 to 31. The first 8 data bits are the data bits for the first byte, and the next 8 bits are the data bits for the second data byte, etc. The 7 check bits are generated by XNORing the participating data bits as indicated by the x's in the table. Each check bit is generated by using a unique pattern of data bits in the data word such that when the check bits are regenerated later and the regenerated check bits compared with the original check bits, a single bit error in any data bit or check bit will be identified uniquely as to its location. In generating check bit 1 all of the bits of data byte 1, i.e. bits 0 through 7 are included, check bit 2 is generated by including all of the data bits in byte 2, i.e. bits 8 through 15, check bit 3 includes all of the eight data bits in byte 3, i.e. data bits 16 through 23, check bit 4 is generated including data bits 24 through 31. It will be apparent to one skilled in the art that using only these data bits which correspond to the data bits in each data word that a parity bit

will be generated for each data word, i.e. that bits 0 through 7 forming check bit 1 constitute a parity bit for byte 1, that data bits 8 through 15 constitute a parity bit for byte 2, data bits 16 through 23 constitute a parity bit for the byte 3 and data bits 24 through 31 constitute a parity bit for the data byte 4.

As per claim 23, Fuoco et al. (herein after referred to as one entity: Fuoco) substantially teaches (Figure 1 and abstract) a computer system and method with add-on memory cards that have error correction code logic on the card, and logic to do partial writes of data words. The system has a central processing unit (CPU), a BUS interconnecting the CPU and the add-on memory cards. The CPU or associated components are configured to write data and read data from the add-on memory as several data bytes constituting data words. The system is further configured either within the CPU or as a separate function to generate parity bits associated with each of the bytes of data the CPU writes to the add-on memory and to read parity bits associated with data the CPU reads from the add-on memory and regenerate new parity bits and compare the newly generated parity bits with the original parity bits to detect data errors on data read from the add-on memory. The system itself does not contain error correction code (ECC). The add-on memory has ECC logic to identify any byte having a single bit error in the data bytes or the parity bits written by the CPU to the add-on memory and to correct all single bit errors in data read from the add-on memory to the CPU. The error correcting code includes logic to generate parity bits in the data bytes written by the CPU to the add-on memory and logic to compare the parity bits written by the CPU with those generated by the error correcting code logic. Fuoco teaches that when writing data by the CPU 10, parity bits are generated for each byte of information written to memory by the parity generating and checking device 13 which also

checks parity on information read from the memory subsystem 22 during a read cycle to determine parity error. The memory controller also provides the necessary signals, such as Row Activation Strobe (RAS), Column Activation Strobe (CAS), Write Enable (WE), Output Enable (OE), and Address (ADDR), etc. to the memory subsystem 22 as shown in FIGS. 1A and 1B. The memory controller reads and writes both data and parity to each of the DRAM cards 26, also as shown in FIGS. 1A and 1B.

Fuoco does not explicitly teach storing the data and parity check bits on a buffer memory as stated in the present application.

However, Fuoco teaches (col. 10 and Figure 7) the RAS and CAS lines are each directed to the input side of OR-gates 80, 82, 84, 86, 88, and 90. The output of the OR-gates 80, 82, 84, 86, 88, and 90 are directed respectively to a series of non-inverting buffers which are minus active enable. The buffers 92, 94, 96, and 98 have inputs from programmable presence detects on the card and the buffers 100 and 102 have grounded input. It is the output from the buffer 100 which is used to detect whether a card is compatible and the output is supplied to one of pins on the SIMM card. The reason for this configuration is that there are only a limited number of pins on an IBM or other SIMM card, and it is the pin on the card which interacts with the output of buffer 100 which configures the system at all cycles other than read or write or refresh output of buffer 100 will be high. The state will signify the card as either a conventional parity card or a card with ECC on board. Furthermore, the Examiner would like to point out that the buffer memory is a memory that is between data input and active storage. The data input being the information from the process 10 and the active storage being the memory subsystem 22. Therefore, if would have been obvious to one of ordinary skill in the art at the time the invention

was made to check the stored data and parity check bits on a buffer memory. This modification would have been obvious to one of ordinary skill in the art because one of ordinary skill in the art would have recognized that by storing the data and parity check bits on a buffer memory the system would improve on the communication between the processor and active storage by reducing the number of possible transmitted errors.

As per claim 24, Fuoco et al. (herein after referred to as one entity: Fuoco) substantially teaches (Figure 1 and abstract) a computer system and method with add-on memory cards that have error correction code logic on the card, and logic to do partial writes of data words. The system has a central processing unit (CPU), a BUS interconnecting the CPU and the add-on memory cards. The CPU or associated components are configured to write data and read data from the add-on memory as several data bytes constituting data words. The system is further configured either within the CPU or as a separate function to generate parity bits associated with each of the bytes of data the CPU writes to the add-on memory and to read parity bits associated with data the CPU reads from the add-on memory and regenerate new parity bits and compare the newly generated parity bits with the original parity bits to detect data errors on data read from the add-on memory. The system itself does not contain error correction code (ECC). The add-on memory has ECC logic to identify any byte having a single bit error in the data bytes or the parity bits written by the CPU to the add-on memory and to correct all single bit errors in data read from the add-on memory to the CPU. The error correcting code includes logic to generate parity bits in the data bytes written by the CPU to the add-on memory and logic to compare the parity bits written by the CPU with those generated by the error correcting code logic. Fuoco teaches that when writing data by the CPU 10, parity bits are generated for each byte of

information written to memory by the parity generating and checking device 13 which also checks parity on information read from the memory subsystem 22 during a read cycle to determine parity error. The memory controller also provides the necessary signals, such as Row Activation Strobe (RAS), Column Activation Strobe (CAS), Write Enable (WE), Output Enable (OE), and Address (ADDR), etc. to the memory subsystem 22 as shown in FIGS. 1A and 1B. The memory controller reads and writes both data and parity to each of the DRAM cards 26, also as shown in FIGS. 1A and 1B.

Fuoco does not explicitly teach storing the data and parity check bits on a buffer memory as stated in the present application.

However, Fuoco teaches (col. 10 and Figure 7) the RAS and CAS lines are each directed to the input side of OR-gates 80, 82, 84, 86, 88, and 90. The output of the OR-gates 80, 82, 84, 86, 88, and 90 are directed respectively to a series of non-inverting buffers which are minus active enable. The buffers 92, 94, 96, and 98 have inputs from programmable presence detects on the card and the buffers 100 and 102 have grounded input. It is the output from the buffer 100 which is used to detect whether a card is compatible and the output is supplied to one of pins on the SIMM card. The reason for this configuration is that there are only a limited number of pins on an IBM or other SIMM card, and it is the pin on the card which interacts with the output of buffer 100 which configures the system at all cycles other than read or write or refresh output of buffer 100 will be high. The state will signify the card as either a conventional parity card or a card with ECC on board. Furthermore, the Examiner would like to point out that the buffer memory is a memory that is between data input and active storage. The data input being the information from the process 10 and the active storage being the memory subsystem 22.

Therefore, if would have been obvious to one of ordinary skill in the art at the time the invention was made to check the stored data and parity check bits on a buffer memory. This modification would have been obvious to one of ordinary skill in the art because one of ordinary skill in the art would have recognized that by storing the data and parity check bits on a buffer memory the system would improve on the communication between the processor and active storage by reducing the number of possible transmitted errors.

As per claim 25, Fuoco et al. (herein after referred to as one entity: Fuoco) substantially teaches (Figure 1 and abstract) a computer system and method with add-on memory cards that have error correction code logic on the card, and logic to do partial writes of data words. The system has a central processing unit (CPU), a BUS interconnecting the CPU and the add-on memory cards. The CPU or associated components are configured to write data and read data from the add-on memory as several data bytes constituting data words. The system is further configured either within the CPU or as a separate function to generate parity bits associated with each of the bytes of data the CPU writes to the add-on memory and to read parity bits associated with data the CPU reads from the add-on memory and regenerate new parity bits and compare the newly generated parity bits with the original parity bits to detect data errors on data read from the add-on memory. The system itself does not contain error correction code (ECC). The add-on memory has ECC logic to identify any byte having a single bit error in the data bytes or the parity bits written by the CPU to the add-on memory and to correct all single bit errors in data read from the add-on memory to the CPU. The error correcting code includes logic to generate parity bits in the data bytes written by the CPU to the add-on memory and logic to compare the parity bits written by the CPU with those generated by the error correcting code logic. Fuoco

teaches that when writing data by the CPU 10, parity bits are generated for each byte of information written to memory by the parity generating and checking device 13 which also checks parity on information read from the memory subsystem 22 during a read cycle to determine parity error. The memory controller also provides the necessary signals, such as Row Activation Strobe (RAS), Column Activation Strobe (CAS), Write Enable (WE), Output Enable (OE), and Address (ADDR), etc. to the memory subsystem 22 as shown in FIGS. 1A and 1B. The memory controller reads and writes both data and parity to each of the DRAM cards 26, also as shown in FIGS. 1A and 1B.

Fuoco does not explicitly teach storing the data and parity check bits on a buffer memory as stated in the present application.

However, Fuoco teaches (col. 10 and Figure 7) the RAS and CAS lines are each directed to the input side of OR-gates 80, 82, 84, 86, 88, and 90. The output of the OR-gates 80, 82, 84, 86, 88, and 90 are directed respectively to a series of non-inverting buffers which are minus active enable. The buffers 92, 94, 96, and 98 have inputs from programmable presence detects on the card and the buffers 100 and 102 have grounded input. It is the output from the buffer 100 which is used to detect whether a card is compatible and the output is supplied to one of pins on the SIMM card. The reason for this configuration is that there are only a limited number of pins on an IBM or other SIMM card, and it is the pin on the card which interacts with the output of buffer 100 which configures the system at all cycles other than read or write or refresh output of buffer 100 will be high. The state will signify the card as either a conventional parity card or a card with ECC on board. Furthermore, the Examiner would like to point out that the buffer memory is a memory that is between data input and active storage. The data input being the

information from the process 10 and the active storage being the memory subsystem 22. Therefore, if would have been obvious to one of ordinary skill in the art at the time the invention was made to check the stored data and parity check bits on a buffer memory. This modification would have been obvious to one of ordinary skill in the art because one of ordinary skill in the art would have recognized that by storing the data and parity check bits on a buffer memory the system would improve on the communication between the processor and active storage by reducing the number of possible transmitted errors.

As per claim 26, Fuoco et al. (herein after referred to as one entity: Fuoco) substantially teaches (Figure 1 and abstract) a computer system and method with add-on memory cards that have error correction code logic on the card, and logic to do partial writes of data words. The system has a central processing unit (CPU), a BUS interconnecting the CPU and the add-on memory cards. The CPU or associated components are configured to write data and read data from the add-on memory as several data bytes constituting data words. The system is further configured either within the CPU or as a separate function to generate parity bits associated with each of the bytes of data the CPU writes to the add-on memory and to read parity bits associated with data the CPU reads from the add-on memory and regenerate new parity bits and compare the newly generated parity bits with the original parity bits to detect data errors on data read from the add-on memory. The system itself does not contain error correction code (ECC). The add-on memory has ECC logic to identify any byte having a single bit error in the data bytes or the parity bits written by the CPU to the add-on memory and to correct all single bit errors in data read from the add-on memory to the CPU. The error correcting code includes logic to generate parity bits in the data bytes written by the CPU to the add-on memory and logic to compare the

parity bits written by the CPU with those generated by the error correcting code logic. Fuoco teaches that when writing data by the CPU 10, parity bits are generated for each byte of information written to memory by the parity generating and checking device 13 which also checks parity on information read from the memory subsystem 22 during a read cycle to determine parity error. The memory controller also provides the necessary signals, such as Row Activation Strobe (RAS), Column Activation Strobe (CAS), Write Enable (WE), Output Enable (OE), and Address (ADDR), etc. to the memory subsystem 22 as shown in FIGS. 1A and 1B. The memory controller reads and writes both data and parity to each of the DRAM cards 26, also as shown in FIGS. 1A and 1B.

Fuoco does not explicitly teach storing the data and parity check bits on a buffer memory as stated in the present application.

However, Fuoco teaches (col. 10 and Figure 7) the RAS and CAS lines are each directed to the input side of OR-gates 80, 82, 84, 86, 88, and 90. The output of the OR-gates 80, 82, 84, 86, 88, and 90 are directed respectively to a series of non-inverting buffers which are minus active enable. The buffers 92, 94, 96, and 98 have inputs from programmable presence detects on the card and the buffers 100 and 102 have grounded input. It is the output from the buffer 100 which is used to detect whether a card is compatible and the output is supplied to one of pins on the SIMM card. The reason for this configuration is that there are only a limited number of pins on an IBM or other SIMM card, and it is the pin on the card which interacts with the output of buffer 100 which configures the system at all cycles other than read or write or refresh output of buffer 100 will be high. The state will signify the card as either a conventional parity card or a card with ECC on board. Furthermore, the Examiner would like to point out that the buffer

memory is a memory that is between data input and active storage. The data input being the information from the process 10 and the active storage being the memory subsystem 22. Therefore, if would have been obvious to one of ordinary skill in the art at the time the invention was made to check the stored data and parity check bits on a buffer memory. This modification would have been obvious to one of ordinary skill in the art because one of ordinary skill in the art would have recognized that by storing the data and parity check bits on a buffer memory the system would improve on the communication between the processor and active storage by reducing the number of possible transmitted errors.

As per claim 27, Fuoco et al. (herein after referred to as one entity: Fuoco) substantially teaches (Figure 1 and abstract) a computer system and method with add-on memory cards that have error correction code logic on the card, and logic to do partial writes of data words. The system has a central processing unit (CPU), a BUS interconnecting the CPU and the add-on memory cards. The CPU or associated components are configured to write data and read data from the add-on memory as several data bytes constituting data words. The system is further configured either within the CPU or as a separate function to generate parity bits associated with each of the bytes of data the CPU writes to the add-on memory and to read parity bits associated with data the CPU reads from the add-on memory and regenerate new parity bits and compare the newly generated parity bits with the original parity bits to detect data errors on data read from the add-on memory. The system itself does not contain error correction code (ECC). The add-on memory has ECC logic to identify any byte having a single bit error in the data bytes or the parity bits written by the CPU to the add-on memory and to correct all single bit errors in data read from the add-on memory to the CPU. The error correcting code includes logic to generate

parity bits in the data bytes written by the CPU to the add-on memory and logic to compare the parity bits written by the CPU with those generated by the error correcting code logic. Fuoco teaches that when writing data by the CPU 10, parity bits are generated for each byte of information written to memory by the parity generating and checking device 13 which also checks parity on information read from the memory subsystem 22 during a read cycle to determine parity error. The memory controller also provides the necessary signals, such as Row Activation Strobe (RAS), Column Activation Strobe (CAS), Write Enable (WE), Output Enable (OE), and Address (ADDR), etc. to the memory subsystem 22 as shown in FIGS. 1A and 1B. The memory controller reads and writes both data and parity to each of the DRAM cards 26, also as shown in FIGS. 1A and 1B.

Fuoco does not explicitly teach storing the data and parity check bits on a buffer memory as stated in the present application.

However, Fuoco teaches (col. 10 and Figure 7) the RAS and CAS lines are each directed to the input side of OR-gates 80, 82, 84, 86, 88, and 90. The output of the OR-gates 80, 82, 84, 86, 88, and 90 are directed respectively to a series of non-inverting buffers which are minus active enable. The buffers 92, 94, 96, and 98 have inputs from programmable presence detects on the card and the buffers 100 and 102 have grounded input. It is the output from the buffer 100 which is used to detect whether a card is compatible and the output is supplied to one of pins on the SIMM card. The reason for this configuration is that there are only a limited number of pins on an IBM or other SIMM card, and it is the pin on the card which interacts with the output of buffer 100 which configures the system at all cycles other than read or write or refresh output of buffer 100 will be high. The state will signify the card as either a conventional parity card or

a card with ECC on board. Furthermore, the Examiner would like to point out that the buffer memory is a memory that is between data input and active storage. The data input being the information from the process 10 and the active storage being the memory subsystem 22. Therefore, if would have been obvious to one of ordinary skill in the art at the time the invention was made to check the stored data and parity check bits on a buffer memory. This modification would have been obvious to one of ordinary skill in the art because one of ordinary skill in the art would have recognized that by storing the data and parity check bits on a buffer memory the system would improve on the communication between the processor and active storage by reducing the number of possible transmitted errors.

Conclusion

This is a Continuation of applicant's earlier Application No. 10608320. All claims are drawn to the same invention claimed in the earlier application and could have been finally rejected on the grounds and art of record in the next Office action if they had been entered in the earlier application. Accordingly, **THIS ACTION IS MADE FINAL** even though it is a first action in this case. See MPEP § 706.07(b). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiries concerning this communication should be directed to the examiner, Mujtaba Chaudry who may be reached at 571-272-3817. The examiner may normally be reached Mon – Thur 6:30 am to 4:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, please contact the examiner's supervisor, Albert DeCady at 571-272-3819.

Mujtaba Chaudry
Art Unit 2133
November 8, 2006

GUY LAMARRE
PRIMARY EXAMINER